Hindawi
Security and Communication Networks
Volume 2021, Article ID 9717747, 14 pages
https://doi.org/10.1155/2021/9717747

Research Article

Anticollusion Attack Strategy Combining Trust Metrics and Secret Sharing for Friendships Protection

Junfeng Tian and Yue Li

School of Cyberspace Security and Computer Institute, Hebei University, Baoding 071000, China

Correspondence should be addressed to Yue Li; 670186807@qq.com

Received 2 May 2021; Accepted 26 June 2021; Published 5 July 2021

Academic Editor: James Ying

Online social networks provide users with services such as online interaction, instant messaging, and information sharing. The friend search engine, a new type of social application, provides users with the service for querying the list of other individuals' friends. Currently, the existing research focuses on independent attacks for friend search engines while ignoring the more complicated collusion attacks, which can expose more friendships that users are not willing to share. Compared with independent attackers, collusion attackers share query results by cooperating with each other. In this article, we propose a resistance strategy against collusion attacks to protect the friendship privacy. The proposed trust metric is based on users' behaviors and is combined with Shamir's secret sharing system, which can transform friendships into secrets. Through secret distribution and reconfiguration, only the participants who meet the query requirements can successfully reconstruct the secret, while the participants who do not meet the query conditions cannot successfully obtain the secret fragments even if they obtain the secret fragments. Experiments are conducted to verify the effectiveness of the proposed strategy and proved that this strategy can greatly limit the number of malicious attackers, greatly reduce the probability of successful collusion attacks, and reduce the number of victims.

1. Introduction

Friendship, as the beginning of social networks, is one of the most important factors in the development of online social networks (OSNs). Friendship is also the basis of social relationships. The friend search engine was born with the development of social networks. It provides a service for users in social networks to browse other users' friends list. According to "findermind," the top 25 friend search engines help users find anyone for free and with high quality [1]. The powerful function of searching friends with friend search engines provides great convenience for users to search for familiar or interested friends and potentially attracts more people to join social networks.

1.1. Problem Identification. Friend search engines may reveal more friendships than users are willing to share, which is considered a privacy violation. Without a proper protection strategy to address such privacy leakage, friendships that users do not want to display are always revealed, which will lead to users no longer using OSNs. Currently, available protection schemes [2] have been shown to resist malicious queries of friendships by independent attackers. The social network can record the query history of each individual requester, and when a query is made to the same user, the attacker always obtains the same friends list as a result of the query. With a defence strategy, an independent attacker cannot query for friendships outside the user's privacy settings.

A complicity attack by multiple queries has emerged [3]. It is accomplished by multiple malicious attackers who share query results by coordinating the query targets and sequences to make users reveal their friendships outside the privacy settings. Collusion attackers can gain access to the users' friendships that cannot be queried by independent attackers. Since existing defence strategies can only protect friendship privacy from independent attacks, they cannot

effectively resist conspiracy attacks. However, the FriendGuard supports only two kinds of friend searches, including unweighted and popularity-based friend search [4]. The methods and characteristics of the conspiracy attacks need to be analyzed, and the query strategy needs to be studied and improved in order to protect the friendship privacy. The data sharing framework can resolve potential data leakage [5]. We focus on the design of an anticollusion attack strategy that is aimed at the privacy of users' friendships in OSNs.

1.2. Methods and Contributions. Friendships serve as the basis for interaction between users in social networks and as an extension of interpersonal relationships in the real world. Collusion attacks on friendships can lead to the leakage of interpersonal relationships outside the user's settings, which will cause a more serious impact on the stability of OSNs. To address such complicity attacks in social networks, we design a privacy protection strategy for friendships that can resist complicity attacks by combining the trust metric [6] with a (t, n) threshold function [7], drawing on the idea of secret sharing. The main contributions can be divided into the following three aspects:

A method to measure trust based on users' interaction behaviors is proposed. Combining the important features of users' interactions, the attributes that affect the trust metric are identified. Direct trust, recommendation trust, and comprehensive trust are calculated, and the friend queriers are classified according to the trust metric to control them when they query friendships.

A privacy-preserving strategy for friendships that can resist conspiracy attacks is proposed. The trust metric is combined with the Shamir Secret Sharing (SSS) system to transform friendships into secrets. The (t, n) threshold function is specially applied so that after sharing the secret, only a subset of the target user's friendships can be successfully queried by satisfying a specific condition, thus protecting the privacy of friendships.

The experimental design and implementation are described. First, the rationality of the trust metric is verified by a probabilistic random function. Then, the security is verified by experimenting against the collusion attack scheme. The feasibility and security are illustrated in terms of the limit rate, the number of victims, the number of attackers, and the probability of successful attacks.

2. Related Works

2.1. Attacks on the Privacy of Friendships. The number of users in OSNs continues to grow. Tens of thousands of users search for new friends and establish new contacts every day. Therefore, the privacy problem in friend search engines has attracted the attention of many researchers. Attacks against the privacy of friendships in OSNs can be divided into two categories: attacks initiated by independent attackers and by colluding attackers.

Regarding independent attacks, research on modeling malicious attacks in OSNs showed that malicious individuals use the actual trust relationship between users and their family and friends to spread malware via OSNs [8]. By changing the display of malicious posts and personal information and hiding him/herself to avoid detection, an attacker in a chameleon attack, which is a new type of deception based on OSNs, is able to destroy users' privacy [9]. Studies have also shown that when the topology of OSNs does not contain cycles, malicious entities will violate users' privacy via active attacks if the network structure is not carefully designed [10]. Due to the rapid development of convolutional neural networks in recent years, applying them to social networks can result in very effective reasoning attacks and make high-precision predictions about private data [11]. In the heuristic attack model based on the Dopv attack [12], the attacker obtains the number of friends of the victim from two published social network graphs by spoofing the trust or browsing the homepage. The tag symmetry attack identifies a pair of friends by marking two fixed-point tags that connect the same edge [13]. An attacker can also identify the friendships of a pair of users by the number of their mutual friends [14]. Although OSN can hide the identity of the user by removing the user's identifier, an attacker can use other contextual information about the OSN to infer the identity of the target user [15].

Collusion attacks involve multiple malicious entities with the aim of launching a malicious attack through the coordination of multiple malicious entities to obtain more private information than is obtained in independent attacks. Multiple malicious entities can be fake accounts that are created by a single attacker or different real attackers [16, 17].

The router and users can maliciously collude to perform a collusion name guessing attack to compromise people's privacy [18]. Compared with independent attacks, collusion attacks are more complex and often exploit system vulnerabilities that independent attacks cannot detect. There is a complex collusion attack strategy in which multiple malicious users coordinate their queries, share the query results, and dynamically adjust their query based on the system's feedback to other malicious requestors [3].

2.2. Protection on the Privacy of Friendships. The actual parameter settings of social network providers have an impact on the display of users' personal information [19]. The personalized privacy measurement algorithm can calculate the user's privacy level, thereby protecting privacy data [20]. Moreover, the classification-anonymity model effectively guarantees the privacy of sensitive data [21]. Users' privacy is secured by encrypting data, and only authorized parties who have obtained the key can decrypt the encrypted content [22]. The blockchain-based secure key management scheme can improve trustworthiness more effectively and efficiently [23].

The additive secret sharing technique can encrypt raw data into two ciphertexts and construct two classes of secure

functions, which can be used to implement a privacy-preserving convolutional neural network [24]. Trust and identity are fundamental issues in social and online environments, and trust management can help users build trust and establish relationships with other users [25]. Existing relationships of users in social networks can be described as one-hop trust relationships, and further multihop trust relationships are built during the recommendation process [26]. When a user involves data items from multiple users, the trust value among users can be used to weigh the weight of user opinions to determine whether the data items are released or not, thus enabling collaborative privacy management [27]. In addition, a series of studies have proposed an unsupervised trust inference algorithm that is based on collaborative filtering in weighted social networks and a fast and robust trust inference algorithm [28, 29] to strengthen the security of social networks via trust inference and to satisfy the goal of differential privacy, a privacy and availability data clustering (PADC) scheme based on k -means algorithm and differential privacy is proposed, which can enhance the selection of the initial center points and the distance calculation method from other points to the center point [30].

However, researchers rarely consider privacy leakage problems caused by the friend search service provided by OSNs. Research on these problems can address the privacy needs of users' friends while ensuring the sociality of OSNs. The solution adopted by most OSNs is to allow each individual user to choose to completely display or completely hide their entire friend list. Moreover, OSNs often default their users to expose the entire friend list, of which most users are unaware [31]. It is conceivable that this setting aims to increase the sociality of the OSN. If users set their friend list to be completely hidden to protect the privacy of their friendships, this setting will substantially affect the sociality of OSNs. There are also some OSNs that set the users' friend list display to "show only a fixed number." For example, on Facebook, the number of friends displayed is set to 8, which limits the flexibility of users in changing their personal settings. However, some researchers have discovered that randomly displaying eight friends is sufficient for third parties to obtain data to estimate friend lists [32]. Moreover, regarding the different privacy settings of users, consider the following example: if A and B are friends, even if user A hides his or her friend list and the requestor cannot query the friend list of A, if user B is set to display his or her friend list, when the requestor queries the friend list of B, the friendships of B and A will be displayed and destroy A's privacy. This problem is referred to as the "mutual effect" [2].

To better protect the privacy of users' friendships in OSNs, a privacy protection strategy in the friend search engine [2] was shown to successfully resist attacks initiated by independent attackers. However, the strategy was unable to defend against collusion attacks initiated by multiple malicious attackers. Subsequently, an advanced collusion attack strategy coordinated by multiple malicious requestors [3] showed that multiple malicious requestors with limited knowledge of OSNs can successfully destroy users' privacy settings in the friend search engine. Another study [33] implemented web applications to detect malicious behavior such as collusion attacks in the friend search engine. However, few researchers have investigated how to resist collusion attacks initiated by malicious attackers in friend search engines.

In this article, we propose an anticollusion attack strategy to fill these research gaps. This strategy distinguishes trusted users from untrusted users based on the credibility among users in OSNs and uses the (t, n) threshold function to limit the querying of requestors in the friend search engine to resist malicious attacks initiated by colluding attackers in OSNs.

3. Collusion Attack Strategy

3.1. Related Definitions. In friend search engines, to strengthen the protection of the user's friendships, a certain number of friendships, such as k, will be displayed when responding to a query request. These k friends are defined as the most influential friends of the users in the OSN. Assume that node N_a exists in the OSN with direct friends $N_{a.i}$ and that set is $F_a^k (i < k)$. Requestor Q_1 wants to query N_a's friendships; two nodes, N_1 and N_2, exist, and $k = 1$. N_1 and N_2 are each user's most important friends.

3.1.1. Unpopular Node. N_a is an unpopular node if nodes $N_{a.i} \in F_a^k$ and $N_a \notin F_k^{a.i}$. As Figure 1 shows, N_0 is an unpopular node.

3.1.2. Popular Node. N_a is a popular node if $N_a \in F_{a.i}^k$ and $N_{a.i} \in F_a^k$. As Figure 2 shows, N_0 is a popular node.

3.1.3. Occupation. If requestor Q_1 queries node N_1, based on the friend search engine display strategy, the query result is $E(N_1, N_2)$. At this time, N_1 has shown his or her most important friend N_2, and N_1 is occupied.

3.1.4. Passive Display. Requestor Q_1 queries the important friend list of N_1. Based on the friend search engine strategy, the query result is $E(N_1, N_2)$. The most important friend who exposes N_2 is N_1, and N_2 is referred to as a passive display.

3.2. Attack Model

3.2.1. Maximum Number of Friends Displayed. Due to the different personal preferences of users in OSNs, their privacy settings will also be different. The maximum number of friends displayed, k, may also be different. This strategy assumes that all nodes have the same k value.

3.2.2. Attackers' Prior Knowledge. Typically, the success of a malicious requestor's attack is closely related to his or her knowledge of OSNs. The attack success rate of malicious requestors who know more about OSNs is expected to be higher. This article assumes that malicious requestors have

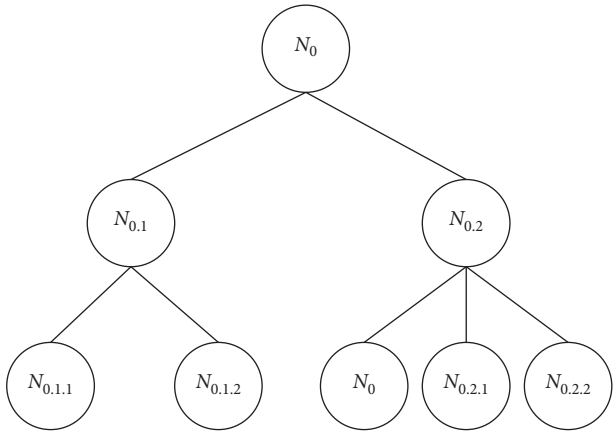

Figure 1: Unpopular node.

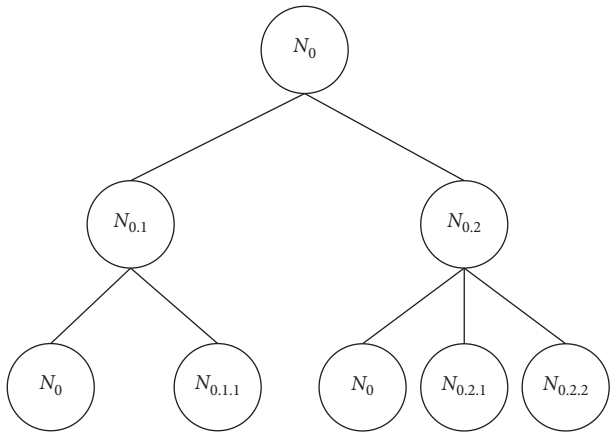

Figure 2: Popular node.

limited knowledge of OSNs and are limited only to target nodes.

3.2.3. Attack Target. The goal of the malicious query is to violate the privacy of the target user in the OSN (i.e., to query the $k+1$th friend of the target node). When the privacy of the target user is set to show a number of friends less than k, the privacy of the target user cannot be violated. Each malicious requestor's attack target is unique, and each collusion attack has only one victim node. Although malicious requestors may infringe the privacy of other users during the query process, only when the privacy of the target node is destroyed is the collusion attack considered successful.

3.2.4. Attack Strategy. Collusion attackers in OSNs can query users' friendships via the friend search engine and query the relationship between users and friends by coordinating the query sequence and query targets.

(1) Attacks on Unpopular Nodes. When the target user is an unpopular node, since there exists at least one node $N_{a.i} \in F_a^k$, and $N_a \notin F_{a.i}^k$. Therefore, the first malicious attacker obtains the set of its friends F_a^k by querying N_a's friends and shares the query result with the new attacker. The new malicious attacker can query N_a's friends $N_{a.i}$ separately by the query result shared by the first malicious attacker and always find the node in $F_{a.i}^k$ where N_a does not exist. Suppose the node is $N_{a.x}$, and a query on $N_{a.x}$ can show its k friends so that it is occupied. At this point, if a query is performed again on the target node N_a, N_a will display its $k+1$th friend $N_{a.(k+1)}$ since $N_{a.x}$ is already occupied.

Suppose there exists a malicious attacker MR_i $(i = 1, 2, \ldots)$, $k = 1$. Taking the unpopular node in Figure 1 as an example to illustrate the attack process on the nonpopular node. The results are shown in Table 1.

(2) Attacks on Popular Nodes. Analogous to the attack on unpopular nodes, the basic idea of the attack on popular nodes is also to expose the $k+1$th friend of the target node by occupying one of its top k friends. However, since both popular nodes and their friends are each other's first k friends, directly querying the friends of the target popular node cannot destroy its friendship privacy by appropriation. Therefore, $N_{a.i}$ is occupied k times by passively displaying $N_{a.i}$. However, when the target node and its friend $N_{a.1}$ are each other's first important friend, they cannot be passively displayed.

Taking N_0 as the target node in Figure 3 as an example, suppose there exist malicious attackers MR_i $(i = 1, 2, \ldots)$ with $k = 3$, The attack process of an attack on popular node N_0 is shown in Table 2.

4. Anticollusion Attack Strategy of Friendships Protection

The collusion attack compromises users' friendship privacy by coordinating the query order through multiple malicious requesters and dynamically adjusting the query target through the query results of others. To solve the problem, we investigate the strategy to resist collusion attacks. In this work, the access control of requestors in the friend search engine is considered, credibility is employed as the restriction condition for requestor queries, and the Shamir Secret Sharing (SSS) system is utilized to control queries.

4.1. Credibility Calculations. In OSNs, the interaction behaviors between users are an important factor that affects the trust metric between users. According to the relationship between users, the trust relationship between two users, i.e., the trust subject and the trust object, can be divided into three types, that are direct trust, recommendation trust, and comprehensive trust. There are four main attributes that are important for credibility calculations.

4.1.1. Number of Interactions. The greater the number of interactions between two users is, the higher the trust between the users is.

4.1.2. Interaction Evaluation. After each interaction, the user gives a corresponding evaluation based on the process, the results, and the importance of the interaction event. The

Table 1: Attack process on unpopular node N_0.

Step	Requestor	Target	Result
1	MR_1	N_0	$E(N_0, N_{0.1})$
2	MR_2	$N_{0.1}$	$E(N_{0.1}, N_{0.1.1})$
		N_0	$E(N_0, N_{0.2})$

evaluation value of the lth interaction is recorded as $C_l \in [0, 1]$.

4.1.3. Interaction Time. Interaction evaluations that are similar to the current time better reflect the user's recent behavior. The closer the evaluation is to the current time, the greater the impact is on direct credibility.

4.1.4. Interaction Events. The weight of the event of the lth interaction between two users is denoted as W_l.

4.1.5. Direct Trust (DT_{ij}). For two user nodes that have historical interactions in the OSN, the credibility of one user to another is referred to as direct trust. A user obtains the credibility evaluation of another based on the historical performance of the user who has interacted with him or her.

If node i and node j have interacted n times in the OSN, after the lth interaction is completed, node i evaluates node j to obtain evaluation value C_l and interaction event weight W_l. Subsequently, the lth interaction time t_l, importance of the lth interaction event W_l, evaluation value C_l of the interaction event of node i with node j, and the influence of the number n of interactions between node i and node j on the evaluation value are considered. The calculation formula of direct trust is expressed as follows:

$$\mathrm{DT}_{ij} = \alpha \cdot \frac{\sum_{l=1}^{n} \Phi(t_l) \cdot C_l \cdot W}{n}, \tag{1}$$

where $\alpha = \sqrt{n/(n+1)}$ is a function of the number of interactions used to adjust the influence of the number of interactions on credibility. The user obtains a high degree of trust only when he or she obtains multiple satisfactory evaluation values. $\varphi(t_l) = \exp(\lceil (t_n - t_l)/T \rceil)$ is the time decay coefficient, where t_n is the nth interaction time (i.e., current interaction time), t_l is the lth interaction time, and T is the time period. The evaluation of an interaction event that is more similar to the current interaction time has a greater impact on credibility. W_l and C_l are the weight of the interaction event between node i and node j and the evaluation value of node i for the event, respectively. This approach can prevent malicious requestors from interacting with the target user by using events with a low weight to gain the trust of the target user while deceiving the user during interaction events with high weights.

4.1.6. Recommended Trust (RT_{ij}). If node i wants to gain a comprehensive understanding of node j, node i needs to obtain the recommended trust for node j via intermediate node c, where node $c = \{c_1, c_2, c_3, \ldots, c_n\}$. The calculation of recommended trust is expressed as follows:

$$\mathrm{RT}_{ij} = \sum_{c=1}^{n} \left(\mathrm{DT}_{ic} \cdot \mathrm{DT}_{cj}\right), \tag{2}$$

where DT_{ic} is the direct trust of user i in user c, DT_{cj} is the direct trust of user c in user j, and the direct trust of user i in user c can be regarded as a recommendation for calculating the recommended trust weights.

4.1.7. Comprehensive Trust (OT_{ij}). The credibility of a user in the OSN must be integrated with his or her direct trust and the recommended trust of other users, which is referred to as comprehensive trust. The weights of direct trust and recommended trust are determined by experimental calculations. In real life, people are generally more inclined to believe their judgments, and the recommendations of others serve only as a reference. Thus, the calculation of comprehensive trust is expressed as follows:

$$\mathrm{OT}_{ij} = u \cdot \mathrm{DT}_{ij} + v \cdot \mathrm{RT}_{ij} \quad (u + v = 1, u > v), \tag{3}$$

where OT_{ij} is the direct trust of node i in node j, RT_{ij} is the recommended trust of node i in node j, and u and v are the weight coefficients of direct trust and recommended trust, respectively.

4.2. Shamir Secret Sharing System. The SSS system is a specific secret sharing scheme designed by Shamir based on language interpolation polynomial theory [34, 35]. This scheme clearly illustrates how to divide data D into n segments so that D can be easily reconstructed from t segments and so that even if all $t - 1$ segments are mastered, D cannot be reconstructed.

In response to collusion attacks in OSNs, this article uses the SSS (t, n) threshold function to control the querying of users' friendships. The (t, n) threshold SSS consists of the following three stages.

4.2.1. System Parameter Setting. n is the number of all participants, t is the threshold, p is a large prime number, and $s \in Z_p$ is the secret to be shared.

4.2.2. Secret Distribution. The secret distributor D chooses a random t degree polynomial.

$$a(x) = s + a_1 x^1 + a_2 x^2 + a_3 x^3 + \cdots + a_{t-1} x^{t-1} \bmod p, \alpha_j \in_R Z_p. \tag{4}$$

The condition $a(0) = s$ is satisfied. D sends $s_i = a(i)$ to participants $P_i, i = 1, 2, \ldots, n$.

4.2.3. Secret Reconstruction. Any number of participants can reconstruct the secret using their secret fragments. Let t participants who want to reconstruct the secret be $P_i, i = 1, 2, \ldots, t$, and let $A = |1, 2, \ldots, t|$.

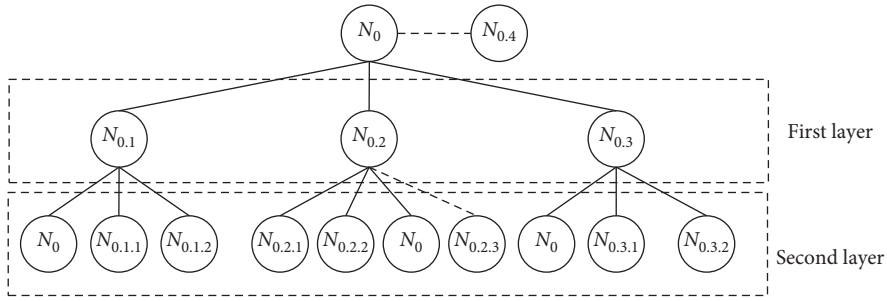

Figure 3: Friendship of popular node N_0.

Table 2: Attack process on popular node N_0.

Step	Requestor	Target	Result
1	MR_1	N_0	$E(N_0, N_{0.1}), E(N_0, N_{0.2}), E(N_0, N_{0.3})$
2	MR_2	$N_{0.2}$	$E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_{0.2.2}), E(N_{0.2}, N_0)$
3	MR_3	$N_{0.2.1}$	$E(N_{0.2.1}, N_{0.2.1.1}), E(N_{0.2.1}, N_{0.2}), E(N_{0.2.1}, N_{0.2.1.2})$
		$N_{0.2.2}$	$E(N_{0.2.2}, N_{0.2.2.1}), E(N_{0.2.2}, N_{0.2.2.2}), E(N_{0.2.2}, N_{0.2.2.3})$
		$N_{0.2}$	$E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_0), E(N_{0.2}, N_{0.2.3})$
4	MR_4	$N_{0.2.3}$	$E(N_{0.2.3}, N_{0.2.3.1}), E(N_{0.2.3}, N_{0.2}), E(N_{0.2.3}, N_{0.2.3.2})$
		$N_{0.2}$	$E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_{0.2.2}), E(N_{0.2}, N_{0.2.3})$
		N_0	$E(N_0, N_{0.1}), E(N_0, N_{0.3}), E(N_0, N_{0.4})$

λ_i is calculated based on the following formula:

$$\lambda_i = \prod_{j \in A/\{i\}} \frac{j}{j-i}. \tag{5}$$

The original secret is restored based on the following formula:

$$s = \sum_{i \in A} s_i \lambda. \tag{6}$$

The security of the SSS depends on the assumption that the parties honestly perform the operations predetermined by the agreement. We consider reliable secret distributors and believe that the administrators of OSNs are honest in the strategy.

4.3. Friend Search Engine with the SSS System

4.3.1. Friendships Transform. When a querier queries the friends of a target user, the friend search engine will return the relationship of edges between nodes among users according to the display strategy. However, according to the SSS system and the requirement of the (t, n) threshold function, the shared secret is $s \in Z_p$ with p being a large prime number. The secret s to be shared in this strategy is the friendship of the target. Therefore, it is necessary to process the representation of an important user's friendship and transform it to the range of Z_p and then share it by the threshold function.

In order to transform the friendships into shareable secrets, we propose a friendship transform algorithm to convert the friendships to satisfy the secret sharing condition. According to the query goal of the querier, the IDs of the first k friends of the target node are first obtained. The friendships transform algorithm is shown as Algorithm 1.

4.3.2. Friendships Protection. In OSNs, users can access the friendships of other users by friend search engines. Multiple malicious requestors can share their query results with each other by coordinating the query target and query sequence, which causes the target user to expose more friends than the user is willing to display. A friend search engine that has introduced the trust metric and SSS can control the queries of users. This control can guarantee that only users whose comprehensive trust reaches the trust threshold can successfully query the friendships of the target user.

Assume that secret distributor D is honest and that each anonymous requestor $P_i, i = 1, 2, \ldots, n$ can obtain a correct secret fragment from D. The number of requestors is higher than the trust threshold for querying the friendships of the target user each time $n_A \geq 2$. The access control process of this solution is described as follows.

Obtain Comprehensive Trust. Requestors $P_i, i = 1, 2, \ldots, n$ request querying the friendships of target user n_a, obtaining comprehensive trust T_{ai} of P_i, and sorting the results in descending order by value based on the interaction between target user n_a and requestor P_i in the OSN.

Classify the Query. Based on trust threshold TR, the requestors are divided into categories A and B. Category A: $T_{ai} \in [TR, 1]$ and category B: $T_{ai} \in [0, TR]$. The number of requestors in the two categories is denoted as n_A and n_B.

Confirm Threshold t. According to the definition of the (t, n) threshold function and the requirements of access control security, requestors who have not reached the

Input: ID_x: ID of the target user
Output: s: the secret to share
(1) Get the IDs of the top k friends of the target node: ID_1, ID_2, ID_3, ..., ID_k;
(2) $SUM_{ID} = \sum_{i=1}^{k} ID_i$;
(3) if SUM_{ID} is prime then
(4) $s = SUM_{ID}$
(5) else
(6) s = find_next_prime(SUM_{ID});
(7) end if

Algorithm 1: Friendships_transform.

system trust threshold cannot successfully query the target user's friendships. Since $T_{ai} < TR$, it is necessary to ensure that requestors in category B cannot successfully query the friendships of the target user. Thus, in each query process, $t = n_B + 1$.

Secret Distribution. The secret distributor D chooses a random t degree polynomial $a(x) = s + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + a_{t-1}x^{t-1} \bmod p, \alpha_j \in_R Z_p$, $a(0) = s$. D sends $s_i = a(i)$ to participants $P_i, i = 1, 2, \ldots, n$.

Secret Reconstruction. n_B requestors in category B, who are arranged in descending order of comprehensive trust, submit the secret fragments s_i obtained in reverse order, and n_A requestors and n_B requestors are divided into n_A groups for secret reconstruction.

Assume that requestor P_i $(i = 1, 2, \ldots, n)$, who queries the friendships of the target user, is arranged in descending order based on the comprehensive trust of the target user n_a. Category A is P_1, P_2, P_3, ..., P_m, and category B is P_{m+1}, P_{m+2}, ..., P_n. Threshold $t = n_B + 1$. As shown in Table 3, category A can be divided into m groups to reconstruct secret s.

The comprehensive trust of the first requestor among the m groups of requestors who participate in the secret reconstruction is greater than the trust threshold set by the target user (i.e., only users trusted by the target user can successfully query the target's friendships). During each secret reconstruction process, the users P_{m+1}, P_{m+2}, ..., P_n who have not reached the comprehensive trust level threshold must submit their secret fragments s_{m+1}, s_{m+2}, ..., s_n obtained from D. Users P_1, P_2, P_3, ..., P_m will submit s_{m+1}, s_{m+2}, ..., s_n. The secret fragment s_i $(i \in [1, m])$ is secretly reconstructed. The threshold $t = n_B + 1$ can ensure that even if P_{m+1}, P_{m+2}, ..., P_n constitute the group of submitted secret fragments, the secret cannot be successfully reconstructed.

4.3.3. Punishment Mechanism. Multiple malicious requestors query the friendships of users by coordinating their query order and query target via the friend search engine. The proposed mechanism further protects the privacy of the target users' friendships by setting the punishment mechanism. When the user who has inquired about the friendships of the target user causes the privacy leak, the comprehensive trust of the inquirers will be reduced, which will make the next query impossible. Assume that before querying the friendships of the target user, the malicious requestors MR_1 and MR_2 are disguised as trusted nodes. If malicious requestors MR_1 and MR_2 have comprehensive trust T_{ai}, $(T_{ai} > T_t)$, during the first query, the malicious requestor MR_1 can successfully reconstruct target node n_a's friendships by secret fragments submitted by category B users and the secret fragments obtained from D. After the malicious requestor MR_1 obtains the query result and shares it with MR_2, malicious requestor MR_2 can require the other nodes based on the query result of MR_1. If the final query result causes the target node to expose the $k + 1$th friend, then the system punishes all nodes that are secretly reconstructed, which reduces the trust value of the user nodes for the reconstructed secret to 1/2 of the original value. The trust decay function is expressed as follows:

$$T_i' = \frac{T_i}{2}. \tag{7}$$

Taking the attack in Section 3.2.3 as an example, assume that the trust threshold is 0.5 and the comprehensive trust of MR_1 and MR_2 is the maximum value of 1. According to the collusion attack strategy, N_3's privacy will be violated.

When the friend search engine detects that the privacy of user N_3 is breached, it will reduce the trust of all users who have queried at this time to punish them. The trust value of MR_2 was originally 1. After the punishment, its comprehensive trust is reduced according to the trust decay function, and the comprehensive trust of malicious requestors MR_1 and MR_2 is reduced from 1 to 0.5. The comprehensive trust obtained from the target user is now lower than the trust threshold, and the next query cannot be performed.

5. Experiment

In this section, we experimentally verify the effectiveness of the proposed anticollusion attack strategy. Our experimental research includes synthetic datasets to verify the validity of the credibility calculations and three large-scale real-world datasets to verify the security of the anticollusion attack strategy.

5.1. Datasets. We generate random numbers that satisfy the previously described conditions of the credibility calculation method, including data on 1000 groups of user interactions,

Table 3: Groups to reconstruct secret s.

Group number	Group member
1	$P_1, P_n, P_{n-1}, \ldots, P_{m+1}$
2	$P_2, P_n, P_{n-1}, \ldots, P_{m+1}$
3	$P_3, P_n, P_{n-1}, \ldots, P_{m+1}$
...	...
m	$P_m, P_n, P_{n-1}, \ldots, P_{m+1}$

and verify the correctness of the trust calculations. In addition, we use three real-world social network datasets to verify the security of the anticollusion attack strategy.

5.1.1. Synthetic Dataset. A random probability function is used to fit users' interactions in OSNs. The setting standards for the time interval of interactions between users and the weights of the interaction events are different for each OSN. We select the interaction data within the time interval ($\Phi(t_l) = 0.367879$) among users in the synthetic dataset. The number of interactions is set to 50; the weights of the interaction events take values in the range [1, 20]; and the interaction evaluation takes values in the range $(0, 1]$ as an example to verify the rationality of the trust calculations, that is, $W_l \in [1, 20]$, $C_l \in (0, 1]$, and $n \in [1, 50]$. The trust between two users may exceed 1 and should be normalized.

5.1.2. Facebook Dataset. In [36], the data from https://Facebook.com capture the friendships among users, which can be modeled as undirected graphs.

5.1.3. Slashdot Dataset. In [37], Slashdot is a technology-related news website and a specific user community, where users can submit and edit news about the current main technology. In 2002, Slashdot launched the Slashdot Zoo function, which enables users to mark each other as friends or enemies. The network establishes links between two friends or enemies among Slashdot users. Therefore, the data in this dataset are directional. This article uses 2009 Slashdot data, and the Slashdot dataset is converted to an undirected graph to reflect users' friendships. Regardless of the direction of the connection between two nodes in the network, an edge is created in the undirected graph for these two nodes.

5.1.4. Gowalla Dataset. In [38], Gowalla is a location-based social networking site in which users share their location by signing in. The friendships collected from Gowalla are undirected. The complete dataset consists of 19, 591 nodes and 950, 327 edges. Due to data size limitations, this program selects only a portion of the data for testing.

We list the main attributes of each dataset in Table 4. The synthetic dataset is used to verify the rationality of the credibility calculations, and the remaining three datasets are used to verify the security of the proposed anticollusion attack strategy.

5.2. Strategy Analysis

5.2.1. Collusion Attack Strategy Analysis. According to the collusion attack model in [3], the collusion attack model has different probabilities of success for collusion attacks on popular and unpopular nodes. The probability of a successful conspiracy attack is mainly related to four factors, such as the degree d of the query node, the number of friends k allowed to be displayed, the layer of the friend relationship tree, and the rank r of the query user among the friends in that layer.

Given a user node of degree d, assume that the probability that one of his friends is ranked among the top k is k/d. Randomly choose the victim node N_0 and one of his top k friends $N_{0.i}$ with degree $d_{0.i}$; then, the probability that N_0 is among the top k friends of $N_{0.i}$ is

$$\begin{cases} \dfrac{k}{d_{0.i}}, & d_{0.i} > k, \\ 1, & d_{0.i} \le k. \end{cases} \tag{8}$$

Simplify it as $\min(k/d_{0.i}, 1)$.

Assuming that the probability of N_0 becoming one of the top k friends of any of its friends is independent, the probability of N_0 becoming a popular node is p. Then, p is denoted by

$$p(N_0) = \prod_{i=1}^{k} \min\left(\frac{k}{d_{0.i}}, 1\right). \tag{9}$$

If N_0 is an unpopular node, the probability of easily destroying the privacy of the target user's friendships by direct query at the first level is

$$p(\text{Attack at layer}_1) = 1 - \prod_{i=1}^{k} \min\left(\frac{k}{d_{0.i}}, 1\right). \tag{10}$$

The number of collusion attackers required is

$$\text{Num}(\textit{Attackers for unpop}) = 1 + k. \tag{11}$$

If N_0 is a popular node, according to the attack flow of compromising the privacy of popular nodes, a malicious attacker cannot directly make N_0 reveal the $k+1$th friend by querying its first layer friends. Therefore, the collusion attackers make the target user N_0's first friend $N_{0.1}$ occupied and thus compromise the target user's friendship privacy by passively displaying it with probability:

$$p(\textit{Attack through } N_{0.1}) = 1 - \prod_{i=1}^{r_{0.1}} \min\left(\frac{k}{d_{0.1.i}}, 1\right), \tag{12}$$

where $r_{0.1}$ is the ranking of N_0 among the friends of $N_{0.1}$.

5.2.2. Anticollusion Attack Strategy Analysis. According to the analysis of the attack success probability of the collusion attack and the total number of malicious attackers required, the probability of successful attack is equation (10), and the

Table 4: Social network dataset property.

Dataset	Synthetic dataset	Facebook	Slashdot	Gowalla
Vertices	1000	63731	82168	196591
Edges	8997	817090	948464	582533
Average degree	—	25.773	12.273	9.668

number of collusion attackers required is $k + 1$. The attack on popular nodes is more complicated. Generally, this attack cannot destroy the privacy of the target user's friendships by querying the first friend only, and the probability of destroying the target user's privacy by occupying the first friend $N_{0.1}$ of the target user is equation (12), where the number of conspiracy attackers required is at least $k + 2$.

In the friendship protection strategy against collusion attacks, the querier first needs to obtain a high level of trust through long-term interaction with the target user, and second, the querier needs to query the friends through the (t, n) threshold function. On the one hand, the anticollusion attack strategy sets a fully trusted querier to help with the query when there are fewer than n queriers. On the other hand, it avoids the situation where all of the malicious queriers have a high trust value.

In the worst case, the number of collusion attackers needed for unpopular nodes is only 2, which requires at least two queries, while the number of collusion attackers needed for popular nodes is 3, which requires at least three queries. When querying by the (t, n) threshold function, the worst case of the class A has $n_A - 1$ malicious attackers among the queriers. The subsequent security analysis will verify and analyze the security of the friendships privacy protection strategy with the worst-case number of malicious attackers against the collusion attack.

5.3. Performance Analysis. In this section, we analyze the rationality of the trust calculations and the security of the anticollusion attack strategy using (t, n) threshold function access control.

5.3.1. Credibility Calculation Rationality. In this article, we propose a trust measure based on the interaction behaviors between users. Considering the number of interactions between two users in a period of time, interaction evaluation, interaction event weight, and other factors, the direct trust degree is calculated by regulating the function. Based on the direct trust degree, the calculation methods of the recommended trust degree and comprehensive trust degree are derived. In this section, the rationality of the trust calculation method is verified by relevant experiments.

As Figure 4 shows, when the time period spanned by user interactions is 2, the time decay coefficient is approximately 0.135; while when the time period spanned by user interactions is 3, the time decay coefficient has dropped to less than 0.1. When the number of user interactions was 9, the ratio of the interaction number conditioning function (INCN) to the number of interactions (IN) was 0.105, while when the number of interactions was 10, the ratio of the interaction number conditioning function to the number of interactions was 0.095. The number of time decays and the ratio of the interaction number conditioning function to the number of interactions were too low to show the more obvious experimental data results. Therefore, the number of interactions between users selected for the experiment ranged from 1 to 9, and the time period spanned by user interactions was selected as 1 or 2.

Based on random numbers, the values of direct trust and recommended trust are calculated by equations (1) and (2), respectively, and the value of the user's comprehensive trust is calculated by equation (3). We selected 1000 sets of data to prove the correctness of the trust calculations. The results are shown in Figure 5.

Figures 5(a)–5(c) show that the results of the direct trust, recommend trust, and comprehensive trust calculations, respectively, are normally distributed. In addition, they are in line with realistic expectations.

5.3.2. Security Analysis. To improve the security and usability of the friend search engine, we assume that OSN administrators can be fully trusted in regard to the friend search engine. When the number of requestors is less than the number of query requests, the administrators can help the requestors complete the query.

In this work, we compare the proposed anticollusion strategy with the original collusion attack. The security of the proposed strategy is verified and analyzed in four aspects, such as limit rate, the number of victims, the number of collusion attackers, and the success rate of the collusion attack. It is assumed that the original conspiracy attacker uses a minimum number of malicious attackers and can compromise the privacy of the target user with a minimum number of queries, and the probability of success of its attack is 1; i.e., the conspiracy attacker can successfully compromise the privacy of the target's friendships in each query.

> *Limit Rate* (LR). The LR of the system is defined as the ratio of the number of users in category B to the number of all users, that is, the proportion of users who cannot successfully query in the friend search engine among all requestors. Based on equation (3) $\mathrm{OT}_{ij} = u \cdot \mathrm{DT}_{ij} + v \cdot \mathrm{RT}_{ij}$ $(u + v = 1, u > v)$, where $\mathrm{DT}_{ij}, \mathrm{RT}_{ij} \in [0, 1]$. The direct trust weight coefficient u is set to 0.6, and the trust threshold is set to 0.5, 0.6, 0.7, 0.8, and 0.9. A total of 1000 experiments are conducted to verify the LR of the proposed strategy.

Figure 6 shows the LR and trust threshold results of the strategy. The value of the direct trust weight coefficient u is 0.6. When the trust threshold is 0.5, the LR of the strategy is approximately 40%. When the trust threshold is 0.6, the LR increases to 80%. At 0.7, the LR increases to almost 100%.

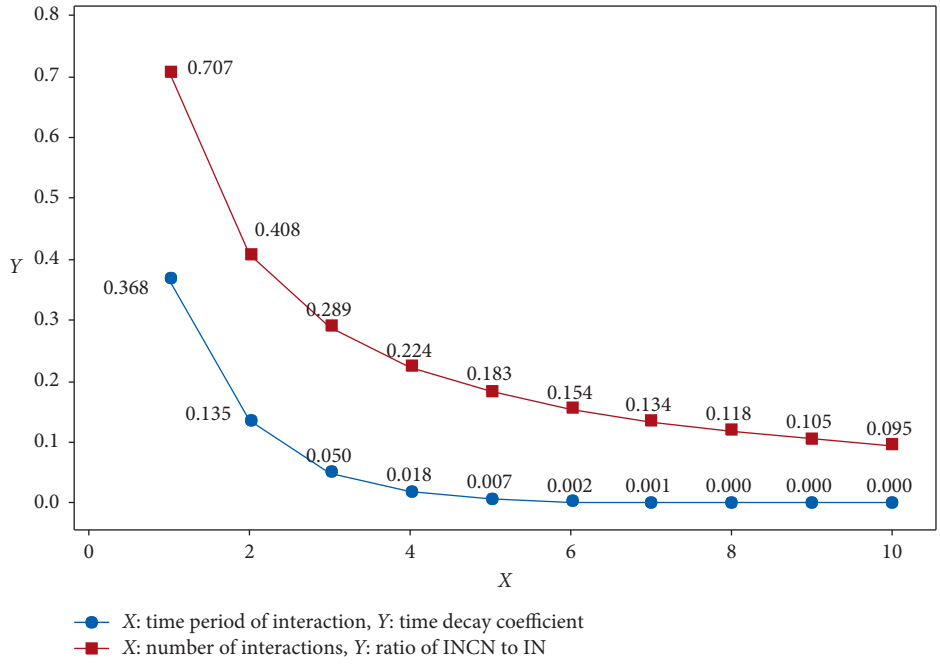

Figure 4: Schematic diagram of data selection.

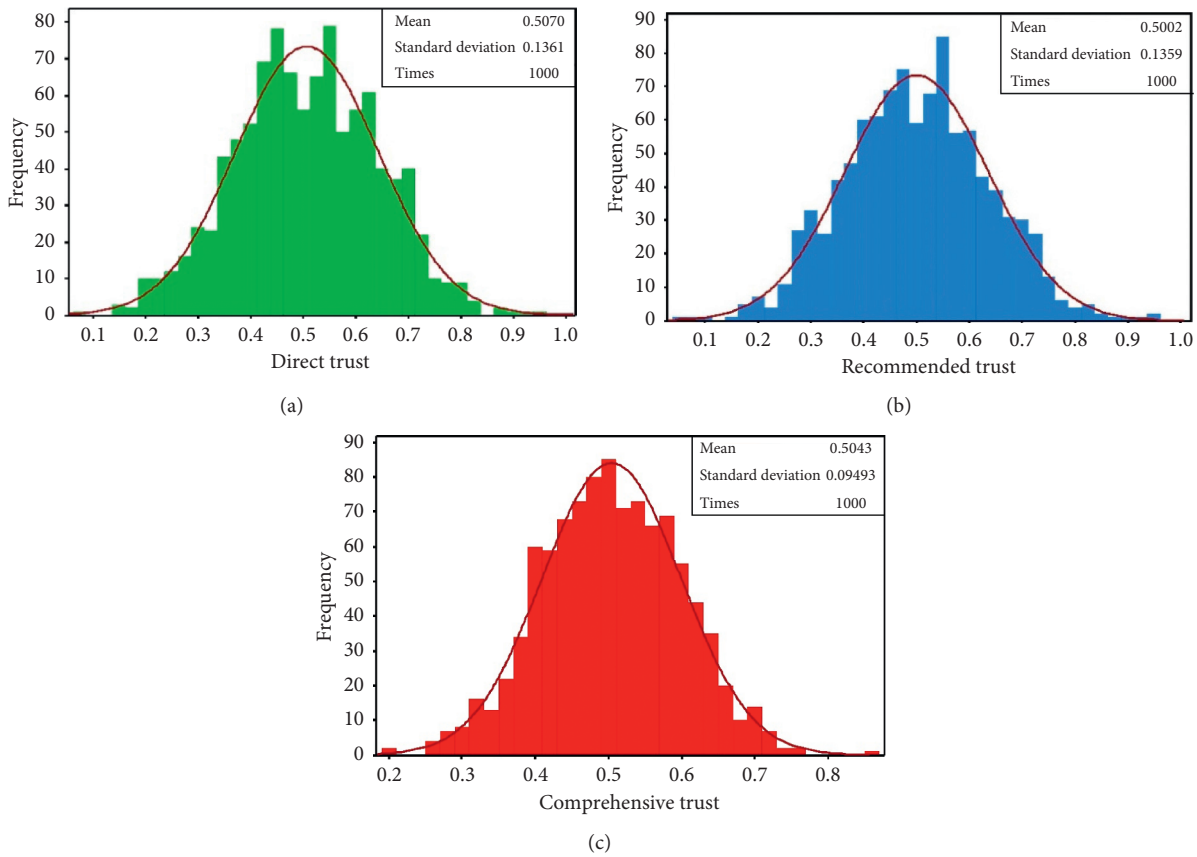

Figure 5: Trust values. (a) Direct trust. (b) Recommend trust. (c) Comprehensive trust.

Therefore, when the trust threshold is 0.7, almost no user reaches the trust threshold, and the friend search engine will not allow any querying. When the trust threshold is 0.6, 80% of users in the OSN cannot reach the threshold. Thus, the number of requestors in the friend search engine is limited, and the safety of the friend search engine is increased.

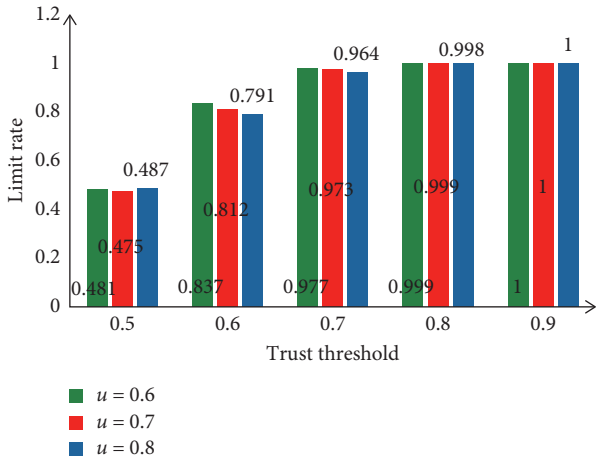

Figure 6: Limit rate under the trust threshold with $u = 0.6$.

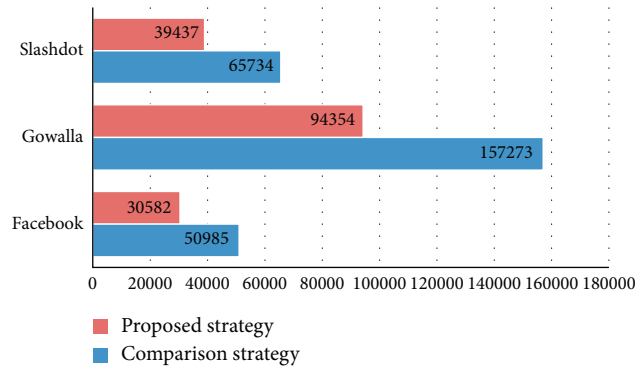

Figure 7: Comparison of the number of victims with a compromised strategy.

Number of Victims. Consider the trust threshold of 0.5 as an example. Sixty percent of users can make normal queries. In the worst case of the friend search engine query, the number of malicious requestors is not limited, and malicious requestors can destroy the privacy of the target user via a one-time collusion attack at the first layer. The probability of successfully destroying the user's privacy is equation (10). The attack can destroy the privacy of 80% of the nodes in OSNs [3].

In a one-time collusion attack, the maximum number of malicious requestors is $n_A - 1$, and the collusion attack performs at least two queries. Thus, the probability of one collusion attack that destroys the target user's privacy at the first layer is

$$\left(\frac{n_A - 1}{n_A}\right)^2 \cdot p\left(Attack\ at\ \text{layer}_1\right). \tag{13}$$

When the trust threshold is set to 0.5 (lowest threshold), 40% of users' queries will be restricted. In this case, the anticollusion attack strategy can reduce the number of users whose privacy is breached by at least 47.9%. Accordingly, the number of users whose privacy is violated decreases. By comparing the Facebook, Gowalla, and Slashdot datasets, we obtain the results shown in Figure 7.

Due to the limitation of the trust threshold, the number of users whose privacy is breached is significantly reduced. The number of users whose privacy is breached in the Facebook and Slashdot datasets is reduced by approximately 20, 000, while the number of users whose privacy is breached in the Gowalla dataset is reduced by approximately 60, 000. In the three datasets, the number of users whose privacy has been violated will be reduced by at least 40%. The proposed strategy greatly reduces the number of users whose privacy is violated, which improves the privacy security of users in OSNs.

Number of Collusion Attackers. Based on the (t, n) threshold function, in the query process of the friend search engine, n inquirers are required to participate in the query, and at least t requestors are required to perform secret reconstruction. Therefore, in a single query process, to ensure that malicious requestors can successfully query, it is necessary to ensure that t requestors are malicious requestors and that the comprehensive trust is higher than the trust threshold. In the best situation, two malicious requestors can destroy the privacy of the target user by making two queries. The total number of attackers required is $2n$, while in the comparison strategy, the number of inquirers required is only 2. Therefore, when the value of n set by the system is larger, more malicious attackers will be needed.

Figure 8 shows that the number of colluding attackers varies with the number of queries n. The number of attackers in the proposed strategy is twice that of the comparison strategy. Under the same conditions, the colluding attackers will need more entities or accounts to make queries with the proposed strategy.

Probability of a Successful Collusion Attack. Assume that malicious requestors who have not interacted with the target user in the OSN want to query the target's friendships. First, excellent long-term interactions with the target are needed to obtain the trust of the target. A successful collusion attack requires multiple malicious requestors to cooperate to coordinate their query order and target, and each malicious requestor can successfully query the friend list of the query target. Therefore, multiple malicious requestors need to maintain excellent interactions with users in the OSN, which will require colluding malicious requestors to spend a substantial amount of time disguising their intentions to obtain the trust of the target user.

Consider the successful collusion attack process in Table 2 as an example. The collusion attack was coordinated by four malicious requestors. MR_1 makes the first requests, and MR_2 determines the target to be queried based on the query results of MR_1. MR_3 queries based on the query result of MR_2 Thus, user $N_{0.2.2}$ will be "occupied," and the new friend $N_{0.2.3}$ of user $N_{0.2}$ can be queried. MR_4 makes a query based on the query result of MR_3 and obtains the $k + 1th$ friend of

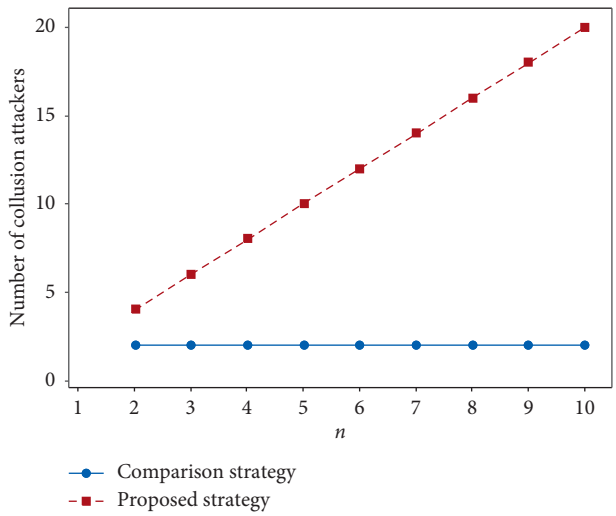

Figure 8: Number of collusion attackers.

N_0, i.e., fourth friend $N_{0.4}$. The privacy of the friendships of user N_0 is destroyed.

Under (t, n) threshold function access control, four malicious requestors, i.e., MR_1, MR_2, MR_3, and MR_4, want to complete this query. First, they need to obtain the high trust of the target nodes, i.e., N_0, $N_{0.2}$, $N_{0.2.1}$, $N_{0.2.2}$, and $N_{0.2.3}$, and all four malicious requestors must have excellent long-term interactions with the target. If a malicious requestor cannot obtain the trust of the target, then $T_{ij} < T_t$, and the previously described attack cannot be successfully carried out. Therefore, a successful malicious attack by colluding attackers requires that all malicious requestors reach the trust threshold.

If malicious requestors already exist in the OSN and have interacted with the target user, this strategy restricts requestors whose trust level is below the trust threshold. A requestor cannot query the target user's friend list under (t, n) threshold function access control. Therefore, when the trust threshold is 0.5, 40% of users who do not reach the trust threshold will not be able to query. As described in the second part of this section, for the collusion attack strategy in [3], if (t, n) threshold function access control is not adopted, the probability that colluding attackers will successfully destroy a user's privacy is 1 for each query. In the (t, n) threshold secret sharing anticollusion attack strategy combined with trust, the comprehensive trust of the requestors who can successfully query the friendships of the target user must be higher than the trust threshold; that is, malicious requestors need to be in category A. Next, we take the trust threshold of 0.5 as an example to discuss the probability that colluding attackers will successfully destroy the privacy of a user's friendships under (t, n) threshold function access control.

If there is a collusion attack, the worst case is that there are enough colluding attackers, and the privacy of the target user is destroyed by just two queries. During a single query, the maximum number of malicious requestors is $n_A - 1$.

For unpopular nodes, the maximum probability of malicious requestors who make two requests is

$$\left[0.6^{(n_A-1)} \cdot \left(\frac{n_A - 1}{n_A}\right)\right]^2. \tag{14}$$

For popular nodes, the maximum probability of malicious requestors who make three requests is

$$\left[0.6^{(n_A-1)} \cdot \left(\frac{n_A - 1}{n_A}\right)\right]^3. \tag{15}$$

In Facebook, Gowalla, and Slashdot, we observe that regardless of whether a popular node or an unpopular node is considered, the number of malicious requestors required to conduct a successful collusion attack can reach 10,000, which is the best case of a successful collusion attack in the three datasets. Therefore, as Figures 9(a) and 9(b) show, in the case of $n_A \geq 2$, when there are at most $n_A - 1$ malicious requestors, the probabilities of successful collusion attacks for unpopular nodes and popular nodes are $p \leq 0.09$ and $p \leq 0.027$, respectively.

Figure 9 shows that when the number of malicious requestors is 2, the anticollusion attack strategy based on (t, n) threshold secret sharing can reduce the probability of a successful collusion attack from 1 to 0.09 and the probability of a successful conspiracy attack on popular nodes from 1 to 0.027. When the number of malicious requestors increases to 18, the anticollusion attack strategy reduces the probability of a successful collusion attack to 0. When the system trust threshold is higher, it is more difficult for malicious requestors to conduct collusion attacks.

Therefore, the trust-based SSS anticollusion attack strategy proposed in this work can substantially reduce the number of users whose privacy is compromised by means of credibility calculations, the trust threshold and the (t, n) threshold function. This strategy restricts user queries based on trust and uses the (t, n) threshold function of the SSS for access control. This strategy can also reduce the probability of successful collusion attacks, which has a significant effect on resisting collusion attacks and can protect the friendship privacy of users in OSNs.

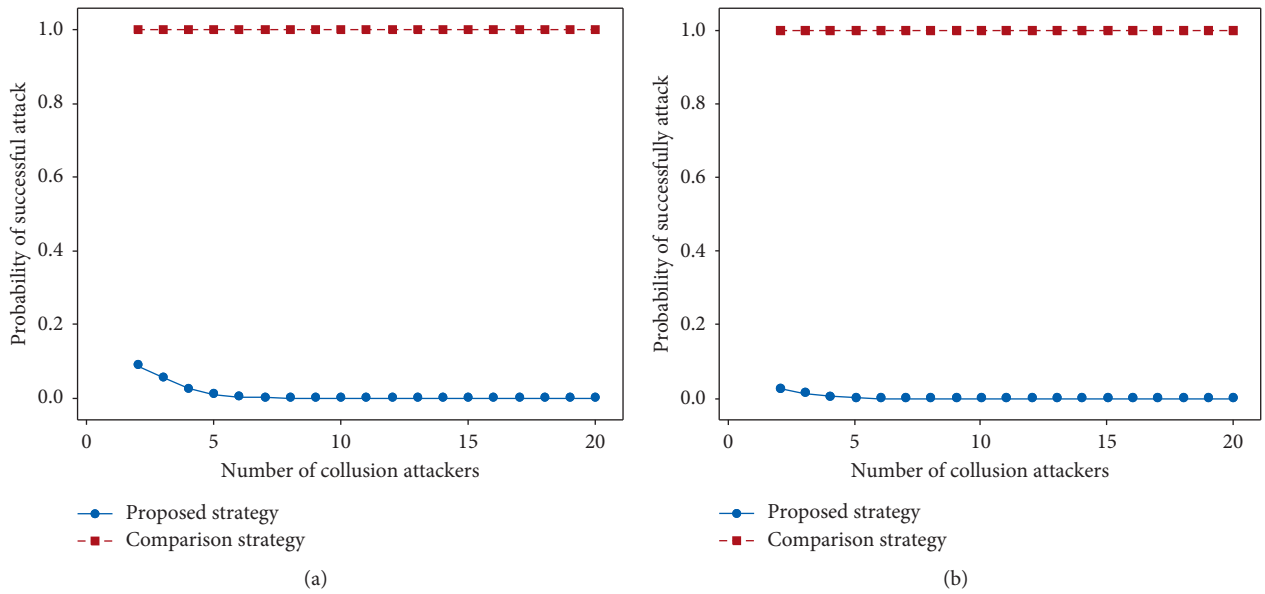

Figure 9: Comparison of the probabilities of a successful collusion attack based on the number of collusion attackers. (a) Unpopular nodes. (b) Popular nodes.

6. Conclusion

To address the problem of collusion attacks that compromise users' friendship privacy, we propose an anticollusion attack strategy that combines the trust metric and (t, n) threshold function. The trust metric is based on the interaction behaviors between users, and the calculation methods of direct trust, recommendation trust, and comprehensive trust are determined by considering the number of interactions, interaction time, interaction evaluation, and event weight. Meanwhile, by converting friendships into secrets and using the (t, n) threshold function to share and reconstruct the secrets, the conspiracy queries of malicious attackers are effectively restricted. The experimental results show that the proposed strategy can significantly reduce the probability of successful conspiracy attacks, reduce the number of victims, and protect the privacy of users' friendships while ensuring normal user queries.

Theoretically, this work simplifies the complex privacy protection of a user's friendships to the user's access control strategy in the friend search engine. This research starts by theoretically analyzing the calculation of trust between two users and applies the (t, n) threshold function to control querying in the friend search engine to protect the privacy of the user's friendships.

Overall, the proposed strategy can successfully decrease the probability of collusion attacks in friend search engines. Specifically, attacking the same number of users requires more attackers, and the number of users who violate the same number of attackers is greatly reduced.

Data Availability

The datasets used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61802106) and the Natural Science Foundation of Hebei Province (F2016201244). The authors of the article would like to express their gratitude to AJE for providing language assistance for this work.

References

[1] finderman: http://www.findermind.com/free-people-search-engines/ , 2021..

[2] L. Na, "Privacy-aware display strategy in friend search," in *Proceedings of the 2014 IEEE International Conference on Communications ICC*, pp. 945–950, Sydney, Australia, June 2014.

[3] L. Yuhong and L. Na, "Retrieving hidden friends: a collusion privacy attack against online friend search engine," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 833–847, 2019.

[4] J. Morris, D. Lin, and A. Squicciarini, "Friendguard: a friend search engine with guaranteed friend exposure degree," in *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, pp. 37–48, Toronto, Canada, June 2019.

[5] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.

[6] T. Junfeng, D. Ruizhong, and C. Hongyun, *Trusted Computing and Trust Management*, Science Press, Beijing, China, 2014.

[7] Q. Weidong, H. Zheng, and L. Xiangxue, *Basics of Cryptographic Protocol*, Higher Education Press, Beijing, China, 2009.

[8] O. Amusan, A. Thompson, T. Aderinola, and B. Alese, "Modelling malicious attack in social networks," *Network and Communication Technologies*, vol. 5, no. 1, Article ID 37, 2020.

[9] A. Elyashar, S. Uziel, A. Paradise, and R. Puzis, "The chameleon attack: manipulating content display in online social media," in *Proceedings of the Web Conference 2020*, vol. 2, pp. 848–859, New York, NY, USA, April 2020.

[10] B. DasGupta, N. Mobasheri, and I. G. Yero, "On analyzing and evaluating privacy measures for social networks under active attack," *Information Sciences*, vol. 473, pp. 87–100, 2019.

[11] B. Mei, Y. Xiao, R. Li, H. Li, X. Cheng, and Y. Sun, "Image and attribute based convolutional neural network inference attacks in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 869–879, 2020.

[12] Y. Fu, W. Wang, H. Fu, W. Yang, and D. Yin, "Privacy preserving social network against dopv attacks," in *Proceedings of the International Conference on Web Information Systems Engineering*, pp. 178–188, Dubai, UAE, November 2018.

[13] C. Liu, D. Yin, H. Li, W. Wang, and W. Yang, "Preserving privacy in social networks against label pair attacks," in *Proceedings of the 12th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2017)*, pp. 381–392, June 2017, https://researchr.org/publication/wasa-2017.

[14] C. Sun, S. Y. Philip, X. Kong et al., "Privacy preserving social network publication against mutual friend attacks," in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops*, pp. 883–890, Los Alamitos, CA, USA, December 2013.

[15] K. S. Min, K. Y. Lee, J. B. Shin et al., "A privacy protection method for social network data against content/degree attacks," *IEICE - Transactions on Info and Systems*, vol. 95, no. 1, pp. 152–160, 2012.

[16] B. Wang, J. Jia, L. Zhang et al., "Structure-based Sybil detection in social networks via local rule-based propagation," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 523–537, 2018.

[17] Q. Zhou and G. Chen, "An efficient victim prediction for Sybil detection in online social network," *IEEE Access*, vol. 8, pp. 123228–123237, 2020.

[18] Z. Xingwen and L. Hui, "Privacy preserving data-sharing scheme in content-centric networks against collusion name guessing attacks," *IEEE Access*, vol. 5, pp. 23182–23189, 2017.

[19] K. Figl and C. Lehrer, "Privacy nudging: how the design of privacy settings affects disclosure in social networks," in *Proceedings of the 28th European Conference on Information Systems (ECIS): A Virtual AIS Conference*, Marrakech, Marocco, June 2020.

[20] Z. Chen, Y. Tian, and C. Peng, "An incentive-compatible rational secret sharing scheme using blockchain and smart contract," *Science China Information Sciences*, vol. 64, no. 10, Article ID 202301, 2021.

[21] J. Xiong, M. Zhao, M. Z. A. Bhuiyan et al., "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.

[22] A. De Salve, R. Di Pietro, P. Mori et al., "A logical key hierarchy based approach to preserve content privacy in decentralized online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 2–21, 2017.

[23] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[24] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[25] V. Kumar and P. Pradhan, "Trust management," *International Journal of Service Science, Management, Engineering, and Technology*, vol. 11, no. 4, pp. 26–44, 2020.

[26] L. Guo, C. Zhang, and Y. Fang, "A trust-based privacy-preserving friend recommendation scheme for online social networks," *Ieee Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 413–427, 2014.

[27] L. Xu, C. Jiang, N. He et al., "Trust-based collaborative privacy management in online social networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 48–60, 2018.

[28] K. Akilal, H. Slimani, and M. Omar, "A robust trust inference algorithm in weighted signed social networks based on collaborative filtering and agreement as a similarity metric," *Journal of Network and Computer Applications*, vol. 126, pp. 123–132, 2019.

[29] K. Akilal, H. Slimani, and M. Omar, "A very fast and robust trust inference algorithm in weighted signed social networks using controversy, eclecticism, and reciprocity," *Computers & Security*, vol. 83, pp. 68–78, 2019.

[30] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.

[31] F. Facebook, "13 million US Facebook users don't change privacy settings," http://www.zdnet.com/article/13-million-us-facebook-users-dontchange-privacy-settings/.

[32] J. Bonneau, J. Anderson, R. Anderson, and F. Stajano, "Eight friends are enough: social graph approximation via public listings," in *Proceedings of the 2nd ACM EuroSys Workshop on Social Network Systems*, pp. 13–18, SNS '09, New York, NY, USA, March 2009.

[33] S. S. Malka, N. Li, and V. M. Doddapaneni, "A web application for studying collusion attacks through friend search engine," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference*, pp. 388–393, Atlanta, GA, USA, 2016.

[34] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[35] E. Dawson and D. Donovan, "The breadth of shamir's secret-sharing scheme," *Computers & Security*, vol. 13, no. 1, pp. 69–78, 1994.

[36] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the SIGCOMM 2009-Proc 2009 SIGCOMM Conf Co-Located Work Proc 2nd ACM Work Online Soc Networks*, pp. 37–42, WOSN 2009, Spain, Barcelona, August 2009.

[37] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[38] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1w082–1090, ACM, San Diego, CA, USA, August 2011.